UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Avaliação de um Sistema de Resposta a Perguntas baseado em LLMs e RAG para um Centro de Informação de Medicamentos

Érika Fernandes Cota

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP
SÃO CARLOS

Érika Fernandes Cota

Avaliação de um Sistema de Resposta a Perguntas baseado em LLMs e RAG para um Centro de Informação de Medicamentos

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientadora: Profa. Dra. Solange Oliveira Rezende

Versão original

São Carlos

2025

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

C843a

Cota, Érika Fernandes
Avaliação de um Sistema de Resposta a Perguntas baseado em LLMs e RAG para um Centro de Informação de Medicamentos / Érika Fernandes Cota ; orientadora Solange Rezende. – São Carlos, 2025.
65 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2025.

1. Processamento de linguagem natural. 2. Agentes conversacionais. 3. LLM. 4. Aprendizado de máquina. 5. Centro de Informação de Medicamentos. 6. RAG. 7. Automação de consultas. 8. Sistemas de Resposta a Perguntas. I. Rezende, Solange, orient. II. Título.

Érika Fernandes Cota

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Original version

São Carlos

2025

Ao Ulisses, minha força e inspiração.

AGRADECIMENTOS

Foram muitas as pessoas que, direta ou indiretamente, me acompanharam neste percurso, e a todas eu agradeço imensamente.

À Solange, meu sincero agradecimento pela oportunidade, pelo apoio, pela generosidade, pela amizade e por todos os aprendizados.

Aos professores do MBA e a toda a equipe do ICMC-USP, obrigada pelos ensinamentos e pelo suporte. Uma menção especial ao professor Ricardo Marcacini, pelas orientações e pelo apoio técnico. Aos colegas de curso, agradeço pelas trocas e compartilhamentos.

Aos colegas e amigos Alexandre Carissimi, Luciana Nedel e Tiago Quim, agradeço por assumirem, apesar das agendas atribuladas, parte da minha carga de trabalho, permitindo que eu pudesse me dedicar ao curso.

Ao William Niemiec, obrigada pela parceria, apoio e inúmeros aprendizados. Ao Douglas Matos, obrigada pela confiança e cooperação.

Aos meus alunos, obrigada pela tolerância com minhas faltas e atrasos nos momentos mais críticos. Em especial, minha gratidão aos petianos, pela paciência, pelo carinho e pelo profissionalismo.

À UFRGS, em especial ao Instituto de Informática (INF), agradeço pelo apoio institucional e pelas oportunidades que me permitem seguir em constante movimento.

Aos meus queridos amigos Chris, Marci, Roberto e Renata, obrigada por estarem sempre por perto!

Por fim, ao meu filho Ulisses e aos meus pais, José Cota e Maristella, peço perdão pelas reiteradas ausências e agradeço por permanecerem sempre ao meu lado.

"The best time to plant a tree was 20 years ago. The second best time is now."

Provérbio Chinês

RESUMO

Cota, E. **Avaliação de um Sistema de Resposta a Perguntas baseado em LLMs e RAG para um Centro de Informação de Medicamentos**. 2025. 65 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

Os Centros de Informação de Medicamentos (CIMs) desempenham um papel crucial ao fornecer informações confiáveis, atualizadas e baseadas em evidências para promover o uso seguro e racional de medicamentos. Uma das principais atividades dos CIMs é a informação passiva (ou reativa), que envolve o recebimento e resposta a questionamentos de profissionais de saúde para subsidiar decisões clínicas. Este trabalho propõe a automação do processo de consulta para reduzir a sobrecarga operacional dos CIMs e permitir que os profissionais se dediquem às atividades de maior valor, como a curadoria e atualização de documentos de referência, além de disponibilizar o serviço em tempo integral. A solução proposta é a criação de um Sistema de Resposta a Perguntas, utilizando Modelos de Linguagem de Grande Escala (LLMs) em conjunto com a arquitetura Retrieval-Augmented Generation (RAG). O objetivo é desenvolver um sistema que possa responder de forma rápida e coerente às consultas, baseando-se em documentos de referência previamente validados. Esta monografia apresenta a implementação e validação de uma Prova de Conceito (PoC) da solução proposta. Utilizou-se um conjunto de documentos de referência fornecidos pelo CIM para gerar uma base textual consolidada. Um prompt detalhado é construído dinamicamente usando técnicas de engenharia de prompt como *role playing* e *chain-of-thought reasoning*. A PoC foi validada usando um conjunto de 56 pares de perguntas e respostas reais. A similaridade semântica entre a resposta gerada pelo LLM e a resposta esperada foi avaliada utilizando outro LLM que gerou um *score* de similaridade variando entre 0 e 100. Seis LLMs distintos foram executados para comparação de desempenho: OpenAI GPT OSS, Gemini 2.5 Flash, Grok 4 Fast, DeepSeek 3.2, Llama 3.1 e GPT5 Mini. O desempenho da solução foi avaliado com base nas pontuações de similaridade obtidas. A pontuação média de similaridade em todos os modelos foi baixa (variando aproximadamente entre 40.95 e 53.75 na validação interna). Além disso, não houve diferença estatisticamente significativa no desempenho entre os modelos avaliados. Os *scores* médios foram menores quando um LLM diferente foi usado para validação. A inspeção manual das respostas revelou uma taxa relativamente alta de respostas do tipo "Não sei" ou similares, levantando a hipótese de que o contexto fornecido ao RAG pode ter ser insuficiente. Concluiu-se que o desempenho médio obtido pela implementação proposta é insuficiente para o domínio da aplicação, não conferindo confiabilidade suficiente. São necessários mais experimentos explorando outros parâmetros, como o tamanho do contexto passado no prompt, entre outros, antes de se assumir o custo de uma validação humana pelos profissionais do CIM.

Palavras-chave: Centro de Informação de Medicamentos (CIM). LLM (Large Language Models). RAG (Retrieval-Augmented Generation). Sistema de Resposta a Perguntas. Processamento de Linguagem Natural (PLN).

ABSTRACT

Cota, E. **Evaluation of a Question-Answering System Based on LLMs and RAG for a Drug Information Center**. 2025. 65 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

The Drug Information Centers (DICs) play a crucial role in providing reliable, up-to-date, and evidence-based information to promote the safe and rational use of medicines. One of the main activities of DICs is passive (or reactive) information service, which involves receiving and responding to inquiries from healthcare professionals to support clinical decision-making. This work proposes automating the inquiry process to reduce DICs' operational workload and allow professionals to focus on higher-value activities, such as curating and updating reference documents, while also enabling the service to be available around the clock. The proposed solution is the creation of a Question Answering System (QAS) using Large Language Models (LLMs) combined with the Retrieval-Augmented Generation (RAG) architecture. The objective is to develop a system capable of providing fast and coherent answers to inquiries, based on previously validated reference documents. This manuscript presents the implementation and validation of a Proof of Concept (PoC) of the proposed solution. A set of reference documents provided by the DIC was used to build a consolidated textual database. A detailed prompt was dynamically constructed using prompt engineering techniques such as role playing and chain-of-thought reasoning. The PoC was validated using a set of 56 real question–answer dataset. The semantic similarity between the answer generated by the LLM and the expected answer was evaluated using another LLM, which produced a similarity score ranging from 0 to 100. Six different LLMs were tested for performance comparison: OpenAI GPT OSS, Gemini 2.5 Flash, Grok 4 Fast, DeepSeek 3.2, Llama 3.1, and GPT5 Mini. The system's performance was assessed based on the obtained similarity scores. The average similarity across all models was low (ranging approximately between 40.95 and 53.75 in internal validation). Moreover, there was no statistically significant difference in performance among the evaluated models. Mean scores were lower when a different LLM was used for validation. Manual inspection of the responses revealed a relatively high rate of "I don't know" or similar answers, suggesting that the context provided to the RAG component might have been insufficient. It was concluded that the average performance achieved by the proposed implementation is inadequate for the application domain, as it does not provide sufficient reliability. Further experiments are needed, exploring parameters such as the size of the context included in the prompt, before assuming the cost of human validation by DICs professionals.

Keywords: Drug Information Centers (DICs). Question Answering System (QAS). Large

Language Models (LLMs). Retrieval-Augmented Generation (RAG) architecture. Natural Language Processing(NLP).

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
BoW	*Bag-of-Words*
CIM	Centros de Informação sobre Medicamentos
GPT	*Generative Pre-Trained Transformer*
IA	Inteligência Artificial
ICL	*In-Context Learning*
LLM	*Large Language Models*
MT	Mineração de Textos
NLG	*Natural Language Generation*
NLU	*Natural Language Understanding*
PLN	Processamento de Linguagem Natural
PoC	Prova de Conceito
RAG	*Retrieval-Augmented Generation*
SBC	Sistemas baseados em Conhecimento
TF	*Term Frequency*
TF-IDF	*Term frequency–inverse document frequency*

SUMÁRIO

1 INTRODUÇÃO

O desenvolvimento de novos fármacos, aliado ao avanço das tecnologias aplicadas à saúde e à crescente disponibilidade de medicamentos, impõe aos profissionais da área a necessidade constante de atualização. Nesse cenário, os Centros de Informação sobre Medicamentos (CIM) desempenham um papel central, fornecendo informações confiáveis, atualizadas e baseadas em evidências, de forma a promover o uso seguro e racional dos medicamentos e contribuir para a efetividade das terapias.

Os CIMs desenvolvem duas atividades principais: informação passiva e informação ativa (Estratégicos, 2020). A informação passiva, também denominada reativa, consiste no recebimento e resposta a questionamentos sobre medicamentos, geralmente oriundos de profissionais de saúde em busca de dados específicos para subsidiar decisões clínicas. Já a informação ativa, ou proativa, envolve a realização de atividades educacionais e a difusão de informações relevantes sobre medicamentos, de forma planejada e sistemática, antecipando demandas e contribuindo para a atualização contínua dos profissionais.

Este trabalho foi desenvolvido a partir da demanda de um CIM que opera em um hospital de referência em Porto Alegre (denominado CIM-POA no restante do texto). O CIM-POA iniciou suas atividades há cerca de 20 anos, com o propósito de apoiar os profissionais da instituição na resolução de dúvidas relacionadas ao uso de medicamentos. Desde então, tem promovido o uso seguro e racional no ambiente hospitalar, consolidando-se como um recurso estratégico na atenção à saúde.

O CIM-POA está integrado à Seção de Farmácia Clínica do hospital e sua atuação abrange desde a etapa de aquisição do medicamento até sua prescrição, dispensação e orientação adequadas, garantindo maior segurança e racionalidade no uso dos fármacos. Esse processo é viabilizado pelo acesso a fontes terciárias atualizadas e a bases de dados fidedignas, que permitem fornecer informações confiáveis em tempo hábil. Assim, o CIM-POA apoia diretamente as equipes médicas e de enfermagem em suas decisões relacionadas a medicamentos, contribuindo para um atendimento mais seguro e qualificado aos pacientes.

O serviço funciona em horário comercial e disponibiliza diferentes canais de atendimento aos profissionais da saúde, incluindo contato telefônico, correio eletrônico e atendimento presencial. As demandas mais frequentes estão relacionadas a aspectos como administração, identificação, posologia, estabilidade e compatibilidade entre medicamentos, evidenciando a abrangência e a importância do serviço prestado.

De acordo com as estatísticas coletadas por diferentes CIM (Estratégicos, 2020), o telefone constitui um dos canais mais utilizados para informação reativa. Segundo a equipe técnica, o uso frequente desse canal se explica pela possibilidade de diálogo ágil entre o

solicitante e o Centro, o que facilita a obtenção das informações mínimas necessárias para a elaboração de respostas mais objetivas e direcionadas. Esses dados reforçam a importância do atendimento síncrono, que possibilita maior interação entre o solicitante e a equipe do CIM, favorecendo a coleta de informações complementares e a produção de respostas mais contextualizadas, conforme preconiza o Ministério da Saúde (Estratégicos, 2020).

Apesar de sua importância estratégica, a manutenção de um CIM operacional enfrenta desafios significativos. A elaboração de respostas a consultas exige a análise criteriosa de múltiplas fontes, o que demanda tempo e esforço por parte da equipe responsável. Além disso, o volume de solicitações combinado com o modelo síncrono de atendimento e a divisão da atenção dos profissionais envolvidos com outras tarefas do setor pode comprometer o tempo de resposta, afetando a agilidade necessária para a tomada de decisão clínica. Manter as bases de conhecimento permanentemente atualizadas e acessíveis, de forma a sustentar respostas rápidas e precisas, configura-se, portanto, como um desafio contínuo.

Este trabalho propõe a automação do processo de consulta como uma alternativa para reduzir a sobrecarga operacional dos CIM. Ao transferir parte das demandas de consulta para um sistema inteligente, é possível liberar os profissionais para atividades de maior valor, como a curadoria, atualização e expansão dos documentos de referência que sustentam o serviço. Dessa forma, garante-se não apenas maior eficiência no atendimento, mas também a qualidade e a atualidade das informações oferecidas aos profissionais de saúde. Tipicamente, a atividade de informação reativa dos CIM está disponível em horário comercial apenas. Um sistema automatizado de consulta permitirá, ainda, a disponibilização do serviço em tempo integral.

Para o desenvolvimento de um sistema automatizado de consulta propõe-se o uso de modelos de linguagem de última geração (do inglês, LLM - *Large Language Models*) em conjunto com a abordagem de *Retrieval-Augmented Generation* (RAG). Vislumbra-se um sistema que possa responder de forma rápida e coerente às consultas recebidas pelo CIM, baseando-se em documentos de referência previamente validados. A proposta busca reduzir o tempo e o esforço necessários para manter o CIM operacional, sem prejudicar o caráter de interatividade, confiabilidade e atualidade das informações disponibilizadas necessários ao serviço. Esta monografia apresenta a implementação de uma prova de conceito (PoC) dessa solução para uso no serviço do CIM-POA[1]. Um dataset de consulta especializado é criado com os documentos de referência utilizados pelos profissionais do CIM-POA. A partir de uma consulta do usuário, o sistema localiza nesse dataset os registros que guardam maior similaridade semântica com a pergunta feita. Em seguida, é montado um

[1] Durante a elaboração deste trabalho, a autora utilizou a ferramenta ChatGPT (versão free) com o objetivo de apoio na codificação e revisão de texto. Após o uso dessa ferramenta, a autora revisou e editou o conteúdo conforme necessário e assume total responsabilidade pelo conteúdo da publicação.

prompt que contém, além da pergunta do usuário, os registros recuperados do dataset como contexto. Esse prompt é fornecido a uma LLM com a orientação de que apenas o contexto disponível no próprio prompt deve ser usado como fonte para geração da resposta. A resposta é coletada e entregue ao usuário.

O objetivo geral desse trabalho é o desenvolvimento de uma PoC que implementa o fluxo descrito acima. Os objetivos específicos são:

- Entender o processo de criação de uma solução com base em LLMs e RAG;
- Identificar os parâmetros que podem e/ou devem ser ajustados nesse tipo de solução para alcançar a confiabilidade desejada;
- Avaliar o desempenho de diferentes modelos de LLM em termos de custo e qualidade da resposta.
- Avaliar a viabilidade de automatizar o processo de informação passiva de um CIM considerando os critérios de confiabilidade e custo.

A avaliação do desempenho dos modelos e da solução como um todo será feita por meio de uma análise experimental que tem como base as seguintes hipóteses:

Hipótese 1: A solução proposta oferece desempenho suficientemente bom (confiabilidade de pelo menos 90%) para o problema proposto

Hipótese 2: Modelos LLM de menor custo e capacidades são suficientes para garantir a confiabilidade desejada.

O texto está organizado da seguinte forma: o Capítulo 2 apresenta os principais conceitos teóricos utilizados ao longo do texto, além de discutir alguns trabalhos relacionados. O Capítulo 3 descreve a metodologia utilizada no desenvolvimento do sistema proposto, incluindo a arquitetura LLM+RAG. O Capítulo 4 detalha a implementação de uma PoC da solução proposta, juntamente com a estratégia de validação utilizada. O Capítulo 5 apresenta e discute os resultados obtidos com a implementação da PoC e o Capítulo 6 finaliza o texto com as conclusões e perspectivas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo estabelece as bases conceituais para a proposta de automação de consultas, iniciando pela discussão de Sistemas baseados em Conhecimento e Aprendizado de Máquina e apresentando a evolução para soluções híbridas que combinam o raciocínio simbólico com a capacidade preditiva do aprendizado de máquina. Em seguida, os conceitos essenciais de Análise e Mineração de Textos são revisados com foco em Processamento de Linguagem Natural e *Large Language Models (LLMs)*. Estratégias de otimização de LLMs para tarefas específicas como o *Instruction Fine-Tuning* e *In-Context Learning* bem como o conceito de engenharia de prompts são também discutidas. Por fim, a arquitetura *Retrieval-Augmented Generation (RAG)* é apresentada como uma extensão do *In-Context Learning* e os Sistemas de Resposta a Perguntas são definidos e relacionados com o uso dos LLMs. A última parte da fundamentação teórica discute outros trabalhos que utilizam LLMs e RAG em contextos clínicos e farmacêuticos.

2.1 Sistemas baseados em Conhecimento e Aprendizado de Máquina

Sistemas baseados em Conhecimento (SBC) são sistemas computacionais que usam conhecimento representado explicitamente para resolver problemas. Eles manipulam conhecimento e informação de forma inteligente e são desenvolvidos para serem usados em problemas que requerem uma quantidade considerável de conhecimento humano e de especialização (Rezende, 2003).

O funcionamento de um SBC baseia-se na utilização de uma base de conhecimento (composta por fatos, regras e heurísticas) e de um mecanismo de inferência, responsável por aplicar essas regras para chegar a conclusões, recomendações ou diagnósticos.

Para desempenhar suas funções de maneira eficaz, um SBC deve ser capaz de interagir com o usuário em uma linguagem clara e de fácil compreensão, de modo a obter as informações necessárias para o processo de inferência. A partir desses dados e do conhecimento previamente incorporado, o sistema desenvolve uma linha de raciocínio que o conduz à formulação de soluções satisfatórias para os problemas apresentados.

Esses sistemas também precisam lidar com informações e regras incompletas, imprecisas ou conflitantes, refletindo a natureza incerta do conhecimento humano. Além disso, devem ser capazes de explicar seu raciocínio, justificando tanto a necessidade de informações externas quanto o caminho lógico que levou às conclusões apresentadas. Essa capacidade explicativa é essencial para garantir a transparência e a confiança do usuário nas decisões do sistema.

Do ponto de vista de desempenho, um SBC deve conviver com eventuais erros,

tal como ocorre com especialistas humanos. Embora possa apresentar falhas pontuais, espera-se que mantenha um nível global de desempenho satisfatório, capaz de compensar possíveis enganos. Em especial, nas situações mais complexas, as soluções fornecidas devem ser equivalentes às produzidas por especialistas humanos, reforçando o papel dos SBC como ferramentas de apoio à decisão e disseminação de conhecimento especializado (Rezende, 2003).

Aprendizado de máquina (AM) (do inglês, *machine learning*) é uma subárea da Inteligência Artificial (IA) voltada para o desenvolvimento de algoritmos e modelos capazes de aprender a partir de dados e melhorar seu desempenho ao longo do tempo, sem a necessidade de reprogramação explícita (Mitchell, 1997). No contexto dos SBC, o aprendizado de máquina desempenha papel fundamental ao contribuir para a atualização, expansão e refinamento da base de conhecimento, tornando o sistema mais adaptável e eficaz diante de novos cenários.

A estrutura da base de conhecimento constitui o núcleo de um SBC e contém o conhecimento específico de determinado domínio, representado de forma explícita por meio de formalismos computacionalmente processáveis. Essa base atua como uma abstração do mundo real, delimitada ao escopo do domínio de aplicação. O formato de representação adotado deve ser compatível com os métodos de manipulação e inferência do sistema, garantindo que o conhecimento possa ser acessado, interpretado e utilizado de maneira eficiente (Rezende, 2003).

O objetivo da representação do conhecimento é fornecer um método sistemático para estruturar e codificar o saber de um domínio, permitindo sua reutilização e expansão (Jakus *et al.*, 2013). Essa abordagem possibilita interpretações variadas e aplicações múltiplas do mesmo conjunto de informações, mantendo a consistência conceitual. Assim, a representação do conhecimento se consolida como um componente essencial tanto para os sistemas baseados em conhecimento quanto para os modelos de aprendizado de máquina, estabelecendo uma ponte entre o raciocínio simbólico e os métodos estatísticos de aprendizagem automática.

A evolução dos SBC e do AM tem conduzido ao desenvolvimento de soluções híbridas capazes de oferecer recomendações personalizadas, contextualizadas e explicáveis. Entre essas aplicações, destacam-se os sistemas do tipo pergunta–resposta, que combinam o raciocínio simbólico dos SBC com a capacidade adaptativa e preditiva dos modelos de aprendizado de máquina (Brunialti *et al.*, 2015).

Nesses sistemas, a interação com o usuário ocorre de forma dialógica e iterativa. O componente simbólico atua na estruturação do conhecimento, representando regras e relações semânticas que garantem coerência lógica e interpretabilidade das recomendações. Já o componente de aprendizado de máquina é responsável por ajustar o sistema a partir dos dados coletados, identificando padrões de comportamento e aprimorando continuamente a

precisão das respostas (Zhao *et al.*, 2024).

2.2 Análise de Textos

Atualmente, dados não estruturados representam pelo menos 80% da informação existente nas organizações (Anandarajan; Hill; Nolan, 2019) e são fontes essenciais para tomada de decisão. Para isso, porém, precisam ser analisados de maneira sistemática para que se transformem em indicadores efetivos que possam ser interpretados corretamente.

O fluxo completo para transformar dados textuais brutos em indicadores analíticos é um processo central da inteligência analítica usando Mineração de Textos (MT). Esse processo pode ser resumido nas seguintes etapas principais (Khurana *et al.*, 2023; Rezende, 2003):

1. Coleta e Preparação Inicial de Dados (Dados Não Estruturados)

   A inteligência analítica lida com uma coleção de textos provenientes de diversas fontes, como notícias, posts em redes sociais, artigos científicos, boletins financeiros, e-mails, planilhas, etc. Tais fontes podem ser externas ou geradas e mantidas pela própria organização. Quando são internas à organização, representam um conhecimento específico cuja análise pode gerar grande diferencial de mercado.

2. Pré-processamento (Transformação para Representação Estruturada)

   Esta etapa visa obter uma representação estruturada dos textos. O pré-processamento de textos envolve diversas técnicas para converter o conteúdo textual não estruturado em uma representação formal(simbólica, numérica ou híbrida) adequada aos objetivos da análise por IA. Como os algoritmos de IA operam sobre dados numéricos, o modelo de espaço vetorial é o mais usado como representação matemática de palavras, sentenças e documentos. Nesse modelo, cada texto é descrito por vetores cujas dimensões refletem características linguísticas ou semânticas, permitindo o uso de técnicas de aprendizado de máquina, medidas de similaridade e modelos estatísticos. Existem também representações simbólicas, amplamente utilizadas em sistemas baseados em conhecimento, que expressam o significado linguístico por meio de regras, ontologias e redes semânticas. Enquanto as representações simbólicas favorecem a interpretação e explicabilidade, as vetoriais se destacam pela eficiência e capacidade de generalização em contextos com grandes volumes de dados.

   O pré-processamento dos textos visando criar o modelo espaço-vetorial inclui:

   a) Construção do Vocabulário: criação de uma lista de palavras únicas da coleção de textos.

   b) Limpeza e Padronização: remoção de números e caracteres especiais, padronização maiúsculas/minúsculas, tratamento de erros de digitação, etc.

c) Refinamento de Termos: remoção de *stopwords*, exclusão de pontuações, pronomes, preposições, artigos e demais palavras não relevantes para a aplicação, simplificação de termos usando técnicas como Radicalização ou Lematização (Anandarajan; Hill; Nolan, 2019).

d) Geração de Atributos: transformação efetiva do documento em um vetor onde cada dimensão/termo é um atributo, o que pode ser feito usando diferentes estratégias como *Bag-of-Words (BoW)* ou *Bag-of-ngrams* entre outras.

e) Ponderação dos Atributos: definir o "peso" de cada atributo para indicar sua relevância. Estratégias mais simples incluem a frequência de ocorrência (*Term Frequency - TF*) ou a TF-IDF (*Term frequency–inverse document frequency*), que pondera a frequência do termo no documento com o inverso da frequência do termo na coleção de documentos. Métodos mais avançados utilizam a representação com *Embeddings* (por exemplo, Word2Vec ou BERT) para lidar com sinônimos e contextualizar palavras, mitigando falhas do BoW. O BERT, por exemplo, gera embeddings representativas de toda a sequência, considerando a ordem das palavras.

3. Extração de Padrões e Modelagem Analítica

Com os dados em uma representação estruturada, o processo avança para a Inteligência Analítica, que tem maior foco em tarefas de análise exploratória da coleção textual. Esta etapa envolve a aplicação de modelos descritivos ou preditivos:

a) Modelos Descritivos ou Extração de Padrões e Análise Exploratória: os modelos descritivos focam na extração de padrões e incluem análise de tópicos relevantes e agrupamento ou *clustering*. Para o agrupamento utilizam-se métricas de similaridade como a similaridade de cosseno (Baeza-Yates; Ribeiro-Neto, 2011) para identificar vizinhos mais similares e gerar redes k-NN (*k-Nearest Neighbors Graph*).

b) Modelos Preditivos ou Classificação: os modelos preditivos realizam previsões ou classificações e incluem classificação de textos, análise de sentimentos e classificação semi-supervisionada.

c) Uso de LLMs (*Large Language Models*): grandes modelos de linguagem podem ser utilizados em tarefas de Inteligência Analítica como sumarização, modelagem de tópicos, análise de sentimentos e classificação.

4. Pós-Processamento e Utilização do Conhecimento

Após a modelagem, segue-se a fase de validação e aplicação. Nesta fase final do fluxo, onde os indicadores analíticos são aplicados na prática, determina-se o desempenho dos modelos por meio de métricas como acurácia e precisão. Nesta fase pode ocorrer também uma análise humana para validação.

2.3 Processamento de Linguagem Natural e LLMs

O Processamento de Linguagem Natural (PLN) é a base do processo de análise de textos. O PLN situa-se na interseção da linguística, ciência da computação e IA, e tem como objetivo principal criar sistemas capazes de compreender, interpretar, manipular e gerar a linguagem humana de maneira útil e significativa (Eisenstein, 2019).

O PLN pode ser classificado em duas grandes áreas (Khurana *et al.*, 2023):

1. Compreensão de Linguagem Natural (do inglês, *Natural Language Understanding – NLU*): envolve a interpretação e compreensão de textos e sentenças em linguagem humana.

2. Geração de Linguagem Natural (do inglês, *Natural Language Generation – NLG*): concentra-se na criação de textos compreensíveis em linguagem natural a partir de dados estruturados ou informações internas do sistema.

Entre as tarefas pesquisadas em PLN, destacam-se a análise e classificação de texto (interpretação do significado, intenção e categorização de textos), a extração e recuperação de informação, a análise da estrutura da linguagem (foco em entender a gramática, a sintaxe e as relações internas do texto) e, por fim, geração de novo conteúdo linguístico ou interação com usuários. Neste trabalho, o foco está no uso de PLN para geração e interação, principalmente nas técnicas que permitem interações conversacionais em sistemas de resposta a perguntas, ou seja, em sistemas que buscam fornecer uma resposta a uma consulta, podendo ser uma resposta extraída do texto original ou envolver a leitura e compreensão do texto.

A evolução das representações de texto, desde o modelo clássico BoW até os LLMs, marca uma progressão no tratamento de dados textuais, migrando de representações esparsas baseadas em frequência para representações densas e contextuais, baseadas em aprendizado de máquina e redes neurais. Os LLMs representam o ápice dessa evolução, sendo desenvolvidos sobre a arquitetura transformer (Vaswani *et al.*, 2023). Exemplos incluem o GPT (*Generative Pre-Trained Transformer* (OpenAI, 2025)), o Meta Llama (Inc., 2025), entre outros. Atualmente, o PLN está fortemente baseado em técnicas de aprendizado de máquina. LLMs têm demonstrado um bom desempenho em várias tarefas de PLN, em especial na implementação de agentes conversacionais.

LLMs são modelos avançados de inteligência artificial especializados no processamento e compreensão de linguagem natural. Eles são baseados na técnica de aprendizado profundo (do inglês, *deep learning*), que permite que o modelo aprenda padrões complexos em grandes volumes de texto. Dessa maneira, esses modelos adquirem a habilidade de modelar aspectos da linguagem humana, como gramática, semântica, entre outros. Os

LLMs são modelos generativos, projetados para gerar textos e pré-treinados com um conjunto massivo de dados textuais. O foco principal durante o pré-treinamento é a predição da próxima palavra. Nesse processo, a rede aprende estruturas linguísticas, associações semânticas e conhecimentos gerais sobre o mundo.

Embora o pré-treinamento permita que o LLM produza textos estruturalmente coerentes, ele pode apresentar limitações para tarefas analíticas específicas ou para responder instruções diretas. Para torná-los mais úteis em aplicações específicas, como sumarização, modelagem de tópicos ou rotulação, é realizada uma etapa adicional denominada *Instruction Fine-Tuning*. Durante o *fine-tuning*, o modelo é treinado com um conjunto de dados mais específico, diretamente relacionado à tarefa-alvo. Essa fase ajusta o modelo para que ele aplique seu conhecimento linguístico geral de forma mais eficaz em um domínio particular. Em geral, o *fine-tuning* requer menos recursos computacionais do que o pré-treinamento pois é feito com uma quantidade muito menor de dados (pelo menos 2 ou 3 ordens de grandeza menor) e envolve apenas o ajuste dos pesos ou de um sub-conjunto dos pesos.

Outra estratégia amplamente utilizada em soluções baseadas em LLM é o *In-Context Learning (ICL)*, que se apoia na capacidade desses modelos de aprender e adaptar-se a partir do contexto fornecido na etapa de inferência, sem necessidade de re-treinamento ou ajuste fino explícito (Brown *et al.*, 2020). Nessa abordagem, os modelos aprendem a partir de exemplos apresentados diretamente no contexto da entrada, em vez de serem re-treinados para cada tarefa específica. Assim, por exemplo, é possível fornecer exemplos de entrada e saída, instruções detalhadas sobre o formato ou estilo da resposta, restrições ou definições de escopo ("considere apenas as informações abaixo..."), entre outras. É importante observar que a qualidade da saída pode variar significativamente conforme os exemplos fornecidos (Zhang *et al.*, 2023) e também com a capacidade do modelo utilizado. Modelos com maior número de parâmetros e treinados em conjuntos de dados mais extensos tendem a apresentar desempenho superior (Kaplan *et al.*, 2020).

Dentro do paradigma de ICL, destacam-se duas formas de uso particularmente relevantes: o *zero-shot learning* e o *few-shot learning* (Brown *et al.*, 2020). No *zero-shot learning*, o modelo é capaz de executar uma tarefa sem ter sido explicitamente treinado ou instruído com exemplos daquela tarefa específica, utilizando apenas o enunciado ou instrução textual como guia. Por exemplo, ao receber o comando "Resuma o seguinte texto", o modelo compreende a tarefa e gera um resumo, mesmo que não tenha sido ajustado especificamente para isso. Já no *few-shot learning*, o modelo recebe alguns exemplos de entrada e saída (geralmente poucos) antes de realizar a tarefa solicitada. Esses exemplos servem como demonstrações contextuais, que ajudam o modelo a ajustar seu comportamento de inferência para produzir respostas mais alinhadas com o padrão esperado.

Essas abordagens são fortemente influenciadas por uma prática denominada engenharia de prompts (do inglês, *prompt engineering*), que consiste em formular instruções e contextos de maneira estratégica para guiar o comportamento do modelo (Liu *et al.*, 2023; White *et al.*, 2023). O *prompt engineering* envolve definir a tarefa de forma clara, estruturar exemplos representativos e aplicar restrições ou estilos de resposta que induzem o modelo a gerar saídas mais precisas, consistentes e relevantes. Em muitos casos, o desempenho de um LLM em ICL depende mais da qualidade e da estrutura do prompt do que da necessidade de qualquer reconfiguração do modelo.

Em especial, as técnicas de *role playing* (interpretação de papéis) (Shanahan; McDonell; Reynolds, 2023) e *chain-of-thought* (cadeia de pensamento) (Wei *et al.*, 2022) são métodos muito utilizados dentro do contexto do *prompt engineering*. Na técnica de *role playing*, o prompt geralmente contém um preâmbulo que define a cena para o diálogo com o LLM e anuncia que o que se segue será um diálogo. Em seguida, tem-se a descrição do papel que o LLM deve interpretar. O prompt inclui ainda alguns exemplos de diálogo no formato padrão, onde as "falas" de cada personagem são indicadas pelo nome relevante seguido por dois pontos (por exemplo, "Usuário:..." "BOT:..."). Ao receber este prompt, o LLM, cujo objetivo é gerar uma continuação que se conforme à distribuição dos dados de treinamento (o vasto corpus de texto humano), gera uma resposta que se alinha com as expectativas de alguém que se encaixa na descrição do preâmbulo. Em essência, o agente de diálogo fará o seu melhor para interpretar o papel do personagem conforme retratado no prompt.

O *chain-of-thought*, por sua vez, é um método simples que explora como a geração de uma cadeia de pensamento (uma série de passos de raciocínio intermediários) melhora significativamente a capacidade dos LLMs de realizar raciocínio complexo. Ele é implementado através do *few-shot prompting*, mas, em vez de simplesmente fornecer exemplos de pares de entrada e saída, o prompt é aumentado com a cadeia de pensamento para a resposta associada. A estrutura do prompt, nesse caso, consiste em triplas: entrada, cadeia de pensamento, saída. A "cadeia de pensamento" é uma série coerente de passos de raciocínio intermediários em linguagem natural que levam à saída final. Essa cadeia imita um processo de pensamento passo a passo, onde o problema complexo é decomposto em etapas menores antes de apresentar a resposta.

Uma extensão importante das estratégias baseadas em ICL é o *Retrieval-Augmented Generation* (RAG), abordagem que combina a capacidade de geração dos LLMs com mecanismos de recuperação de informações externas. O objetivo do RAG é suprir a limitação de conhecimento estático dos modelos, permitindo que eles acessem dados atualizados ou especializados no momento da inferência (Lewis *et al.*, 2020).

No RAG, antes de o modelo gerar uma resposta, um módulo de recuperação semântica busca informações relevantes em fontes externas como bases documentais,

artigos científicos, relatórios institucionais ou bancos de dados vetorizados. Os trechos recuperados são inseridos no prompt como contexto adicional, e o modelo aplica ICL sobre esse material, gerando uma resposta informada pelas fontes fornecidas. Dessa forma, o RAG não altera os pesos do modelo, apenas aumenta o contexto de entrada, permitindo um processo mais flexível e menos custoso. Essa arquitetura tem sido amplamente utilizada em sistemas de busca inteligentes, chatbots corporativos, assistentes de pesquisa e aplicações científicas, nas quais é essencial restringir a geração de respostas às fontes de conhecimento verificáveis (Lewis *et al.*, 2020).

Na implementação básica da arquitetura RAG, que tem sido referenciada pelo termo *naive RAG*, os documentos recuperados de uma base de conhecimento são simplesmente concatenados ao prompt, sem etapas adicionais de filtragem, reordenação ou verificação de consistência, ou seja, sem um controle de qualidade sobre esse contexto que será fornecido ao LLM. As evoluções mais recentes dos sistemas RAG incorporam técnicas de re-ranking semântico, fusão de evidências, mecanismos de verificação factual (Izacard; Grave, 2021; Ram *et al.*, 2023), entre outras, resultando em soluções mais robustas, interpretáveis e adaptáveis a diferentes domínios de aplicação.

2.4 Sistemas de Resposta a Perguntas

Os Sistemas de Resposta a Perguntas (do inglês, QAS - *Question-answering systems*) podem ser definidos como um processo automático capaz de entender questões formuladas em linguagem natural e de responderem com a informação solicitada (Mollá-Aliod; Vicedo, 2010).

O objetivo fundamental de um QAS é fornecer uma resposta clara à consulta do usuário, encontrando a resposta mais relevante em uma única iteração. A resposta pode ser obtida por meio de extração direta de um segmento de um documento ou pela geração de uma síntese coerente a partir de fragmentos textuais (M. Zhang W.E., 2022).

QAS podem ser classificados com base no tipo de tarefa que visam resolver (Biancofiore *et al.*, 2024):

***Open-Goal QA*:** Responde a perguntas explorando texto não estruturado (por exemplo, contextos de fóruns de discussão) ou agrupando perguntas em classes predefinidas, como FAQs.

Factoid QA : Responde a fatos específicos (por exemplo, “Quem é Leonardo Di Caprio?”).

Visual QA: Gera respostas que descrevem o conteúdo de uma imagem sobre a qual a pergunta é feita, buscando uma resposta correta consistente com o conteúdo visual.

Os avanços em PLN e ML alavancaram o uso de QASs nos últimos anos. Em especial, os LLMS permitem não apenas gerar respostas em um amplo espectro de consultas, mas, principalmente, aprimorar a interação dinâmica e conversacional com os usuários.

2.5 Trabalhos Relacionados

Um estudo recente (Silva; Gomes, 2025) propôs o desenvolvimento de um assistente de suporte à decisão clínica baseado em LLMs integrados à abordagem de RAG, com foco na seleção de medicação antidepressiva. A solução combina documentos clínicos específicos do paciente (histórico, folhetos dos medicamentos e queixas atuais) com modelos de linguagem generativos, permitindo a geração de recomendações personalizadas.

A solução foi projetada para execução em infraestrutura local, utilizando LLMs genéricos e de código aberto, visando a redução de riscos de privacidade e facilidade para substituições do modelo generativo.

A estratégia de avaliação da solução utilizou 10 modelos (comerciais e abertos) em duas configurações: query (consulta única) e chat (interação contínua). O desempenho dos modelos foi avaliado segundo cinco critérios: (i) consideração correta do contexto, (ii) acerto na medicação, (iii) explicação e fundamentação da resposta, (iv) qualidade textual e (v) aderência ao formato solicitado.

Os resultados da avaliação para 40 casos clínicos mostraram diferenças importantes no desempenho total entre os modelos e as duas configurações de consulta. Considerando todos os modelos utilizados, as respostas às perguntas melhoraram com o uso da configuração query (consulta única), ou seja, de modo geral, os LLMs tiveram um desempenho melhor em um único prompt ao analisar os dados. Adicionalmente, os modelos comerciais apresentaram melhor desempenho geral, especialmente em clareza e estrutura das respostas, enquanto modelos abertos se destacaram em aspectos específicos, como contextualização e precisão na escolha da medicação.

Zhou *et al.* (2025) propõem uma abordagem baseada em LLMs e RAG para aumentar a eficiência e reduzir o custo do controle de qualidade de rótulos farmacêuticos, um processo tradicionalmente manual e demorado que depende de especialistas para validar cada informação presente nos rótulos de medicamentos. A confiabilidade (minimização das alucinações) e a rastreabilidade do processo são os requisitos principais do problema.

A arquitetura do sistema proposto é composta por quatro módulos principais: (i) um banco de dados hierárquico que organiza tabelas de estudos clínicos segundo compostos, indicações e protocolos; (ii) um sintetizador de consultas que decompõe o texto do rótulo em declarações discretas e gera consultas contextualizadas; (iii) um módulo de recuperação hierárquica que utiliza busca em profundidade e reranking baseado em LLM para selecionar as tabelas mais relevantes; e (iv) um verificador de autoconsistência, responsável por validar

cada valor textual contra as evidências recuperadas.

A solução proposta tem como objetivo principal apoiar especialistas, oferecendo verificações automáticas e destacando pontos que requerem revisão humana. Assim, a partir de um parágrafo do rótulo a ser verificado e um contexto (informações sobre composto, indicação terapêutica e protocolo clínico) fornecidos pelo especialista, o sistema busca por evidências relevantes dentro do banco de dados hierárquico. Em seguida, com base em técnicas de engenharia de prompt e *few-shot learning*, o LLM analisa o texto do rótulo e o decompõe em declarações discretas, transformando cada uma delas em consultas estruturadas que representam as unidades de informação a serem verificadas. Essas consultas são complementadas com os elementos contextuais fornecidos na etapa anterior, garantindo maior precisão na correspondência com as tabelas de ensaios clínicos. As consultas são então processadas pelo módulo de recuperação hierárquica, que executa uma busca em profundidade no banco de dados hierárquico de tabelas de estudos clínicos. Esse banco organiza as informações em uma estrutura em árvore, agrupando os dados por composto, indicação e protocolo. O sistema recupera múltiplas tabelas candidatas e, em seguida, aplica um mecanismo de reranking baseado em LLM, que prioriza as tabelas mais relevantes para cada consulta. Uma vez recuperadas as tabelas mais prováveis, a verificação de autoconsistência é feita. Nesse estágio, cada afirmação extraída do rótulo é transformada em uma pergunta e comparada com os valores presentes nas tabelas correspondentes. O sistema confirma se os dados textuais (como dosagens, amostras ou resultados) são consistentes com as evidências do ensaio clínico. Caso existam múltiplas fontes possíveis, o sistema consolida as respostas com base na consistência interna das evidências. Após a verificação, os resultados são apresentados para o especialista de forma visualmente intuitiva. O texto analisado é exibido com códigos de cores que indicam o grau de verificação. Essa codificação visual permite que o usuário identifique rapidamente quais trechos exigem revisão manual, aumentando a eficiência do processo de controle de qualidade. Por fim, o especialista pode revisar as passagens sinalizadas, ajustar as consultas ou fornecer realimentação sobre a correspondência das evidências. Esse retorno pode ser utilizado para aprimorar o sistema de prompting e o reranking, contribuindo para a melhoria contínua do pipeline.

A estratégia de validação da solução baseou-se em um estudo de caso com dados farmacêuticos reais, estruturado para avaliar a precisão e a consistência do sistema. Foi definido um conjunto de referência (*gold standard*) a partir de rótulos oficiais de medicamentos e criado um cenário de teste balanceado com exemplos reais e sintéticos. A verificação das informações utilizou uma abordagem de autoconsistência baseada em LLMs, voltada a garantir rastreabilidade e confiabilidade nas respostas. Por fim, o desempenho do pipeline foi comparado a abordagens de referência por meio de métricas como sensibilidade, especificidade e acurácia.

Os resultados experimentais demonstraram alta precisão e robustez do pipeline proposto, que alcançou 96,1% de acurácia em validações internas, superando abordagens tradicionais de Naïve RAG e modelos baseados apenas em LLMs (como o GPT-4). Com o uso apenas da tabela mais relevante recuperada (Top-1), o sistema manteve 92% de acurácia e 100% de especificidade, evidenciando a eficácia da arquitetura hierárquica proposta.

3 PROPOSTA DE SOLUÇÃO

Propõe-se a criação de um QAS para apoiar as atividades de informação reativa em CIMs, reduzindo a demanda de trabalho da equipe, que hoje precisa ficar de plantão para esse atendimento, e liberando-a para as atividades essenciais de curadoria e atualização dos materiais de consulta e referência.

A solução proposta é baseada em LLMs visando disponibilizar uma interação mais natural com o usuário, mimetizando o atendimento que é feito tradicionalmente por telefone. Além disso a arquitetura RAG é utilizada para que as fontes de consulta do sistema de recuperação de informações sejam aquelas definidas pelo CIM.

Durante a fase de concepção da solução, foram consideradas alternativas mais tradicionais de recuperação de informação, como o uso exclusivo de *embeddings* vetoriais combinados a medidas de similaridade (por exemplo, cosseno ou TF-IDF). Essas abordagens são eficientes para tarefas de busca direta, permitindo identificar documentos ou trechos relevantes em função da proximidade semântica com a consulta do usuário. No entanto, elas se limitam a recuperar trechos brutos, sem capacidade de interpretar nuances linguísticas, integrar múltiplas fontes ou adaptar o nível de detalhamento da resposta conforme o contexto da pergunta. Por outro lado, a adoção de um modelo baseado em LLMs em conjunto com a técnica RAG oferece vantagens significativas nesse sentido: o agente pode dialogar de forma natural e contextualizada, sintetizando informações de diferentes documentos e apresentando respostas estruturadas, claras e ajustadas ao perfil do interlocutor. Assim, a escolha por uma arquitetura apoiada em LLMs+RAG justifica-se pela necessidade de interação conversacional confiável, com interpretação semântica adequada e capacidade explicativa, aspectos essenciais para um sistema de suporte em ambiente clínico-hospitalar.

Neste trabalho, uma prova de conceito (PoC) de um QAS voltado ao CIM-POA é apresentada. O objetivo é automatizar parte do processo de resposta a consultas sobre preparo, conservação e administração de medicamentos, utilizando técnicas de IA baseadas em LLMs combinadas com RAG. A Figura 1 apresenta a lógica da solução proposta. A partir de um conjunto de documentos-base, estrutura-se uma base de consulta com dados codificados em um modelo de *embeddings* semânticos. Cada consulta feita ao sistema é codificada utilizando o mesmo modelo de *embedding*, permitindo que seja feita uma busca semântica na base pelos documentos mais relevantes para os termos da consulta. Esses documentos são incorporados a um modelo de prompt para instanciar um prompt com contexto expandido que contém a pergunta feita pelo usuário e um contexto de busca confiável. Esse prompt é enviado para uma LLM para gerar a resposta à pergunta do usuário a partir apenas das informações passadas como contexto.

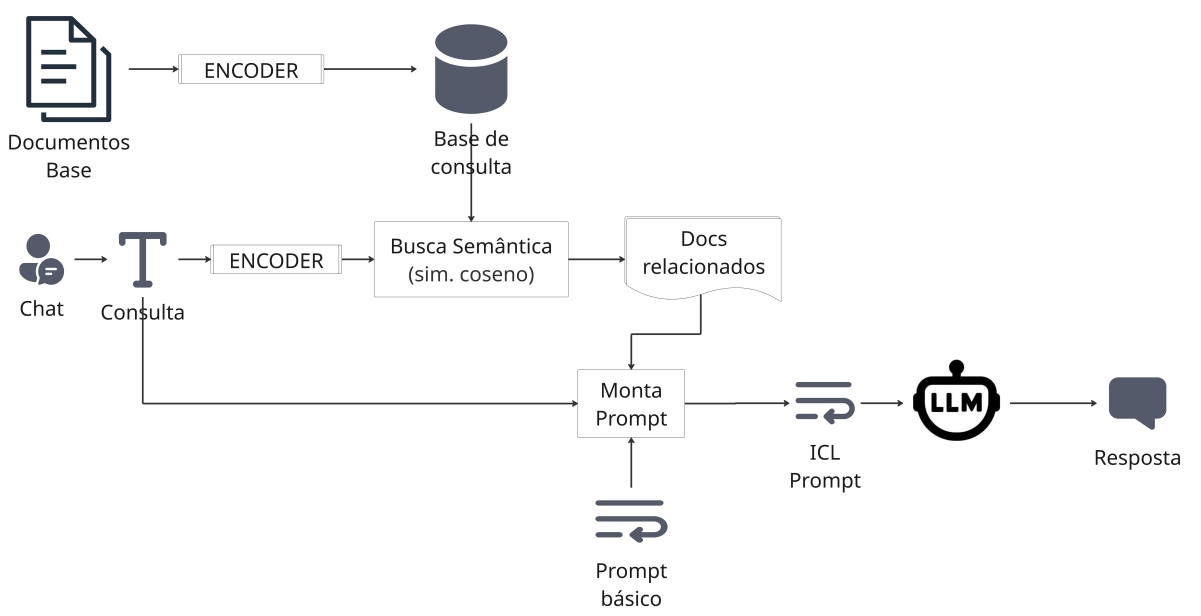

Figura 1 – Arquitetura da PoC proposta

Fonte: Elaborado pela autora.

Para o desenvolvimento da solução, são necessários três blocos principais, a saber:

1. Ingestão e Processamento de Dados
2. Indexação e Armazenamento
3. Recuperação e Geração de Respostas (RAG)

A metodologia prevê ainda uma etapa de validação da PoC com a execução de um conjunto de perguntas-teste (representando consultas reais ao CIM) e a análise qualitativa das respostas geradas pelo agente. Essas etapas serão detalhadas nas seções seguintes.

3.1 Ingestão e Processamento de Dados

Essa etapa é responsável pela construção do dataset de consulta especializada a partir da leitura dos materiais de referência técnica.

O CIM, por definição, realiza uma curadoria na literatura científica e desenvolve materiais que são utilizados tanto para as atividades de informação ativa quanto passiva. Assim, há um conjunto de documentos que já são utilizados como base de consulta para os profissionais que atuam no atendimento aos usuários do serviço. No caso do CIM-POA, estão disponíveis os seguintes documentos de referência:

Tabela de medicamentos injetáveis: documento de 97 páginas em formato PDF. O conteúdo é dividido entre informações referentes a pacientes adultos (45 páginas) e pacientes pediátricos/neonatos (47 páginas). Para cada medicamento, a tabela contém informações sobre a apresentação, reconstituição, vias de administração, solução para diluição, volume usual da diluição, concentração máxima da solução, estabilidade, compatibilidade com PVC, necessidade de equipamento fotossensível e observações adicionais. Junto ao nome do medicamento, pode haver, ainda, uma indicação visual relativa ao custo, ajuste pela função renal e risco de ototoxicidade. Por fim, algumas linhas da tabela são sinalizadas em cor amarela ou vermelha e com indicação visual de Alta Vigilância[1].

Tabelas de Estabilidade: documento em formato Excel composto de 7 planilhas (Uso tópico, Antissépticos desinfetantes, Soluções orais, Preparações Oftalmológicas, Insulinas UBS, Soluções multidoses UBS, Produtos de uso tópico UBS). As tabelas incluem informações sobre a apresentação, estabilidade, validade e conservação do produto após abertura e observações.

Histórico de perguntas e respostas: junto aos documentos de consulta, foram fornecidas duas planilhas com registro de perguntas recebidas pelo CIM-POA e as respostas elaboradas pelo serviço. Os registros foram feitos em dois períodos diferentes (antes de julho de 2025 e entre julho e setembro de 2025). Os documentos contêm questões sobre a administração, preparo e estabilidade de diversos fármacos. As perguntas, frequentemente feitas por técnicos de enfermagem e enfermeiros focam em informações práticas, como diluições mínimas e máximas, tempos de infusão, vias de administração e compatibilidade entre medicamentos e soluções. Além disso, há várias entradas que abordam a estabilidade de produtos após a abertura ou preparo, como a validade de colírios, géis e soluções diluídas. Uma dessas planilhas será usada como parte da documentação de referência e a outra será usada na etapa de validação.

Esses documentos são processados com o auxílio de bibliotecas como PyPDFLoader e CSVLoader (do ecossistema LangChain), para gerar uma base textual consolidada.

3.2 Indexação e Armazenamento

O conteúdo é segmentado em *chunks* e transformado em vetores de *embeddings*, possibilitando a busca semântica eficiente posteriormente. A segmentação em *chunks* foi o único pré-processamento feito nos dados. Outros pré-processamentos, como limpeza, padronização ou refinamento dos termos, não foram aplicados, pois os LLMs já são

[1] Neste trabalho, essa sinalização não será considerada, pois não impacta a qualidade da resposta dentro dos objetivos estabelecidos para a PoC.

treinados para lidar com as variações que esses pré-processamentos corrigem. Além disso, o modelo de *embedding* utilizado é do tipo semântico, que também considera o contexto da frase.

Os *embeddings* gerados são armazenados em formato .parquet, garantindo persistência e reaproveitamento entre execuções. Essa etapa evita o reprocessamento desnecessário e permite o uso incremental de novos documentos de referência.

3.3 Recuperação e Geração de Respostas (RAG)

A etapa de recuperação constitui o núcleo inteligente da arquitetura baseada em RAG, responsável por conectar a base de conhecimento institucional à capacidade de geração textual do modelo de linguagem. Nessa etapa, o sistema interpreta a pergunta do usuário e realiza uma busca semântica nos *embeddings* previamente construídos a partir dos documentos de referência do CIM, identificando os trechos mais relevantes e contextualmente adequados. Esses fragmentos são então incorporados a um modelo de prompt previamente definido, criando-se um prompt concreto que é enviado ao LLM, garantindo que a resposta gerada esteja fundamentada em fontes oficiais e atualizadas. Essa integração entre recuperação e geração permite reduzir o risco de alucinações do modelo, assegurando maior precisão, rastreabilidade e confiabilidade das informações fornecidas.

A identificação dos documentos de referência mais relevantes para cada consulta é feita com base na métrica de similaridade de cosseno. Gera-se um *embedding* da pergunta usando o mesmo modelo utilizado para os documentos de referência. No momento da consulta, calcula-se a similaridade de cosseno entre os *embeddings* da pergunta e da referência e selecionam-se os documentos com maior similaridade para serem passados como contexto ao LLM.

A construção do prompt básico utiliza técnicas consolidadas de *prompt engineering*, em especial a técnica de *role playing*, a fim de garantir clareza, contexto e consistência para o agente baseado em LLM.

3.4 Estratégia de Validação

Uma estratégia de validação da PoC foi definida para avaliar de forma objetiva a precisão e a confiabilidade do agente baseado em RAG. Para isso, utilizar-se-á um conjunto de perguntas e respostas reais, coletadas do histórico de consultas do CIM-POA, anotadas em um arquivo estruturado, e que não é usado como material de consulta no RAG.

Cada pergunta do conjunto de teste é enviada ao agente, que processa a consulta por meio da camada de recuperação e gera uma resposta via LLM, fundamentada nos documentos de referência. A resposta gerada pelo LLM é então comparada com a resposta de referência (presente no arquivo fornecido pelo CIM-POA).

Foram consideradas duas formas para se realizar a comparação da resposta de referência com aquela gerada pelo LLM. A primeira possibilidade é utilizar a métrica de similaridade de cosseno. Nesse caso, as duas respostas devem passar por um pré-processamento para minimizar variações que afetam a comparação, mas não representam diferenças semânticas de fato. Em seguida, devem ser gerados os respectivos *embeddings* e então, a similaridade pode ser medida.

A segunda alternativa consiste em utilizar também um LLM para a comparação. Nesse caso, constrói-se um modelo de prompt que recebe as duas respostas como parâmetro e solicita ao LLM que faça a avaliação da similaridade semântica entre elas, emitindo, por exemplo, um *score* de similaridade e justificando essa nota.

Considerando o contexto de aplicação, avaliou-se que a alternativa baseada em LLM seria a mais adequada, pois é possível que haja uma boa variação tanto no texto da resposta esperada quanto na resposta gerada sem, no entanto, alterar a semântica da mensagem. Como mencionado, LLMs lidam naturalmente com essas variações, conferindo um resultado potencialmente mais realista.

Há, contudo, um risco potencial de viés quando a análise de similaridade é feita pelo mesmo modelo que gerou a resposta. Para lidar com esse risco, serão feitas duas medidas de similaridade utilizando dois modelos distintos, o modelo utilizado na geração da resposta e um segundo modelo como "contraprova".

Com base nas medidas de similaridade obtidas para todas as perguntas do conjunto de teste, calculam-se estatísticas descritivas, como média, mediana, desvio padrão, mínimo e máximo. Esses indicadores fornecem informações sobre o desempenho global do sistema, como o nível geral de concordância semântica do agente com as respostas reais, a existência de inconsistências ou casos em que o agente pode gerar respostas menos confiáveis ou mesmo a identificação de consultas que apresentam maior dificuldade ou para as quais a resposta da LLM diverge significativamente da referência.

Além disso, considerando a grande variedade de LLMs disponíveis e suas diferentes capacidades, o agente é executado com diferentes modelos para fins de comparação de desempenho.

4 IMPLEMENTAÇÃO DA PROVA DE CONCEITO

Este capítulo detalha a implementação do protótipo de agente de consulta farmacêutica, desenvolvido em um ambiente Colab Notebook seguindo a metodologia descrita no Capítulo 3. A implementação foi estruturada em etapas que serão descritas na sequência.

4.1 Configuração do Ambiente e Dependências

O ambiente foi configurado em Google Colab, garantindo acesso a recursos de computação e integração com o Google Drive para armazenamento dos documentos-base. Além das bibliotecas típicas para manipulação de dataframes e outros recursos de manipulação de dados, as seguintes bibliotecas Python foram utilizadas:

LangChain e LangChain Community: Para orquestrar o fluxo RAG, incluindo carregamento de documentos, divisão de texto e interação com o modelo de linguagem.

Sentence-Transformers: Para gerar os embeddings vetoriais dos textos.

Pandas: Para a manipulação e estruturação dos dados carregados, especialmente dos arquivos Excel.

PyPDF: Para o carregamento e extração de texto de documentos em formato PDF.

LangChain-OpenAI e Scikit-learn: Para interagir com o modelo de linguagem via API e para calcular a similaridade de cosseno, respectivamente.

Para acesso aos LLMs foi utilizado o OpenRouter[1], uma plataforma de acesso unificado a modelos de linguagem criada para permitir que desenvolvedores e usuários utilizem vários modelos de IA por meio de uma única API e interface padronizada.

4.2 Carga e Pré-processamento da Base de Conhecimento

A base de conhecimento do QAS foi construída a partir dos múltiplos documentos de referência fornecidos pelo CIM-POA, armazenados no Google Drive.

O documento PDF (Tabelas Injetáveis) foi carregado usando a biblioteca PyPDFLoader. Essa biblioteca extrai o texto do PDF, organiza em partes (denominadas ”documentos”) e fornece metadados como número de página, caminho do arquivo e posição no texto.

[1] https://openrouter.ai/

As sete planilhas contidas no arquivo Excel foram lidas uma a uma considerando as pequenas diferenças no número e título das colunas de cada tabela. Cada linha das planilhas presentes naquele arquivo foi convertida em um documento de texto estruturado, formatando as colunas em uma sentença descritiva para fornecer contexto semântico ao LLM. Foram incluídos ainda metadados como o nome do arquivo, o índice da linha e o rótulo da aba.

As perguntas e respostas utilizadas como parte do dataset de referência estão contidas em uma planilha Excel com três colunas: medicamento, pergunta e resposta. O texto das perguntas e respostas é resumido e contém apenas as informações essenciais anotadas por um profissional do serviço. Para enriquecer o dataset, foi gerado, com o auxílio da ferramenta NotebookLM[2], um arquivo JSON contendo um conjunto estruturado de perguntas e respostas formuladas como frases completas. Esse arquivo JSON foi lido e adicionado à base de conhecimento da solução.

Ao final desta etapa, todos os textos extraídos (97 páginas do PDF, 216 registros das planilhas e 371 registros de perguntas e respostas) foram consolidados em uma lista única, totalizando 684 documentos.

4.3 Vetorização e Indexação (Chunking e Embeddings)

Para otimizar o processo de recuperação de informação, a base de conhecimento consolidada foi processada da seguinte forma:

Divisão de Documentos (*Chunking*): os documentos foram divididos em fragmentos menores (*chunks*) de até 500 caracteres, com uma sobreposição de 50 caracteres entre eles. Essa técnica garante que a informação contextual não seja perdida nas fronteiras dos fragmentos e permite que a busca por similaridade seja mais granular e precisa. O processo resultou em 1.173 chunks.

Geração de *Embeddings*: cada *chunk* de texto foi então convertido em um vetor numérico (*embedding*) por meio do modelo *paraphrase-multilingual-mpnet-base-v2* (Reimers; Gurevych, 2019). Esse modelo foi escolhido por ser bem referenciado em contextos de múltiplos idiomas e considerando que a documentação de referência contém termos em português e inglês. Alguns modelos classificados como da área da farmácia foram avaliados, mas eram muito específicos e o foco neste trabalho não são os medicamentos em si, mas a resposta adequada a uma dúvida.

Armazenamento: Os textos dos *chunks*, juntamente com seus metadados de origem e os *embeddings* gerados, foram salvos em um arquivo de formato Parquet. Com isso,

[2] https://notebooklm.google.com/

evita-se a necessidade de reprocessar os documentos e gerar os embeddings a cada execução do sistema, economizando tempo e recursos computacionais.

4.4 Lógica de Recuperação e Geração

O agente RAG propriamente dito foi implementado utilizando uma estratégia clássica (*naive*) através de uma função principal que executa os seguintes passos a cada nova pergunta do usuário:

1. Busca Semântica: a pergunta do usuário é primeiramente convertida em um vetor de *embedding* usando o mesmo modelo *paraphrase-multilingual-mpnet-base-v2*.

2. Cálculo de Similaridade: o vetor com a pergunta é comparado com todos os *embeddings* dos *chunks* armazenados na base de conhecimento. A métrica de similaridade de cosseno é utilizada para calcular a "distância" semântica entre a pergunta e cada *chunk*.

3. Recuperação de Contexto: os 3 *chunks* com a maior pontuação de similaridade são selecionados como o contexto mais relevante para responder à pergunta.

4. Construção do Prompt: um prompt detalhado é montado dinamicamente a partir do prompt básico mostrado na Figura 2. Ele instrui a LLM a atuar como um farmacêutico clínico, a basear sua resposta exclusivamente no contexto recuperado, a usar uma linguagem clara para profissionais de saúde e a citar a fonte da informação. A pergunta do usuário e os textos dos *chunks* recuperados são inseridos neste modelo de prompt.

5. Geração da Resposta: o prompt final é enviado ao modelo LLM através da API do OpenRouter. O modelo então gera uma resposta em linguagem natural, seguindo as instruções e utilizando as informações fornecidas.

Como mencionado, a solução proposta foi executada com seis LLMs distintas, visando identificar diferenças no desempenho em função da capacidade do modelo utilizado. O Capítulo 5 apresenta os detalhes sobre esses resultados.

4.5 Teste e Avaliação de Desempenho

As seguintes hipóteses foram objeto da análise experimental implementada:

Hipótese 1: A solução proposta oferece desempenho suficientemente bom (confiabilidade de pelo menos 90%) para o problema proposto

```python
def build_prompt(query, docs):
    context = "\n\n".join(docs["text"].tolist())
    modelo_prompt = f"""
      Você é um farmacêutico muito bem qualificado, especializado em
      farmacologia clínica.
      Seu papel é responder a dúvidas de outros profissionais de saúde
      (enfermeiros, técnicos em enfermagem, médicos, entre outros) sobre
      questões relacionadas a administração,
      preparo, diluição e estabilidade de diversos medicamentos, além de
      questões sobre interações medicamentosas ou outras relacionadas à
      farmacologia clínica.
      Para responder às perguntas, você deve se basear apenas no contexto
      fornecido.
      Indique, ao final da resposta, o documento usado como fonte
      da informação.
      Se não houver informação suficiente, diga que não sabe, não tente
      inventar uma resposta.
      Utilize uma linguagem que possa ser entendida por outros
      profissionais da saúde que não são farmacêuticos, em especial
      enfermeiros e técnicos de enfermagem. Símbolos devem ser seguidos
      por seus significados.

      ### Contexto:
      {context}

      ### Pergunta:
      {query}

      ### Resposta:
      """
    return modelo_prompt
```

Figura 2 – Modelo de prompt do agente de consulta para CIM

Fonte: Elaborado pela autora.

Hipótese 2: Modelos LLM de menor custo e capacidades são suficientes para garantir a confiabilidade desejada.

Para validar essas hipóteses e avaliar a eficácia da PoC, foi definida a estratégia ilustrada nas Figura 3. Utilizou-se um conjunto de 56 pares de perguntas e respostas que foram registradas pelo CIM-POA entre os meses de julho e setembro de 2025 e não foram usadas como parte do material de referência para o RAG. Esse conteúdo foi originalmente disponibilizado em formato de planilha Excel, contendo as colunas ”medicamento”, ”pergunta”, ”resposta” e ”tema”. Assim como foi feito para o material de consulta, essa planilha foi transformada, usando o NotebookLM, em um arquivo JSON

estruturado com os pares de pergunta e resposta formatados como frases completas.

Cada uma das 56 perguntas do dataset de avaliação foi submetida ao agente para gerar uma resposta. A similaridade semântica entre as respostas geradas e as respostas esperadas foi avaliada utilizando também os recursos dos LLMs. Para isso, foi utilizado o modelo de prompt apresentado na Figura 4. O prompt recebe as duas respostas e retorna um *score* de similaridade, juntamente com uma justificativa. A escolha pela língua inglesa para este prompt visa minimizar a influência do prompt na escolha dos LLMs e nos resultados, uma vez que muitos LLMs ainda são treinados a partir de datasets nessa língua.

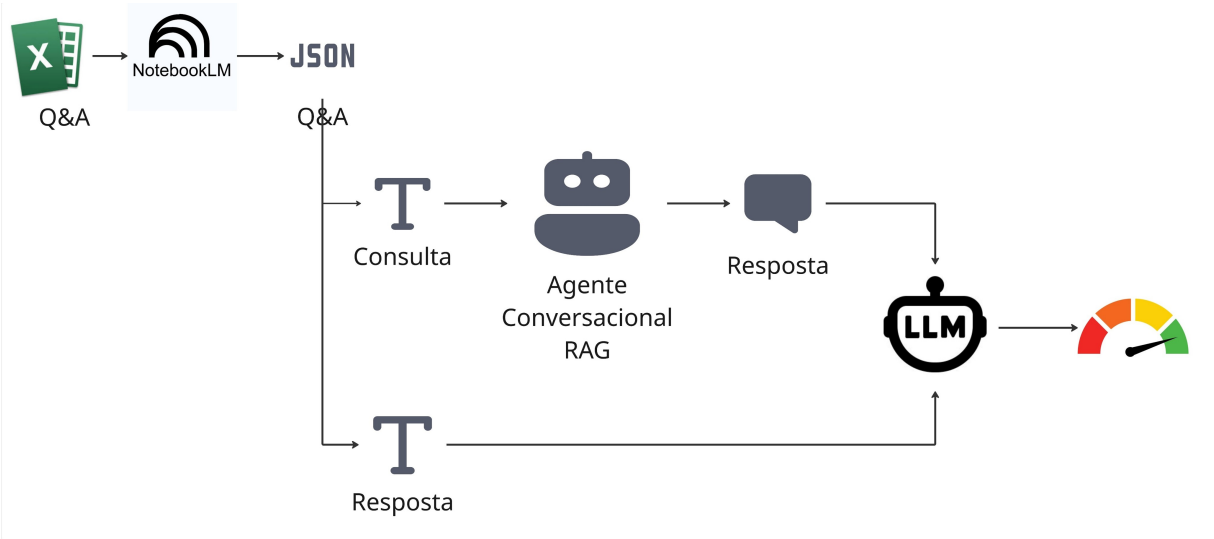

Figura 3 – Estratégia de Avaliação da PoC

Fonte: Elaborado pela autora.

O procedimento de validação foi executado duas vezes para cada LLM avaliado. Uma validação utilizou o mesmo modelo para geração da resposta e avaliação da similaridade com a resposta esperada. A segunda validação utilizou um LLM distinto na etapa de avaliação de similaridade. Todas as pontuações de similaridade resultantes foram coletadas para análise estatística (média, mediana e desvio padrão) e os resultados são detalhados no Capítulo 5.

```python
def build_prompt_Aval(text_a, text_b):
    promptAval = f"""
      Compare the two texts below and provide a semantic similarity score from 0 to 100,
      where 0 means completely unrelated and 100 means semantically identical.
      Also, briefly explain your reasoning.

      Text A: "{text_a}"
      Text B: "{text_b}"

      Return your answer in JSON format exactly like this:

      {{
       "similarity_score": <number>,
       "explanation": "<brief explanation>"
      }}

      Do not include any text outside the JSON object. Return only valid JSON.
    """
    return promptAval
```

Figura 4 – Modelo de prompt do procedimento de validação

Fonte: Elaborado pela autora.

5 RESULTADOS EXPERIMENTAIS

Este capítulo apresenta os resultados da avaliação experimental proposta na Seção 4.5. Inicialmente, são descritos os seis LLMs escolhidos. Em seguida, os resultados obtidos por cada um são apresentados e discutidos, incluindo as limitações e ameaças à validade desse estudo.

5.1 LLMs Utilizadas nos Experimentos

Foram escolhidos seis LLMs para esta avaliação: OpenAI GPT OSS, Gemini 2.5 Flash, Grok 4 Fast, DeepSeek 3.2, Llama 3.1 e GPT5-Mini. A Tabela 1 apresenta as principais informações sobre cada modelo.

O GPT-OSS-20b [1] é um modelo de 21 bilhões de parâmetros de peso aberto (*open-weight*) lançado pela OpenAI sob a licença Apache 2.0. O modelo foi treinado no formato de resposta Harmony da OpenAI e oferece suporte à configuração de nível de raciocínio, ajuste fino (*fine-tuning*) e capacidades de agente, incluindo chamada de funções e saídas estruturadas.

O Gemini 2.5 Flash-Lite[2] é um modelo leve de raciocínio da família Gemini 2.5, otimizado para latência ultrabaixa e alta eficiência de custo. Segundo o provedor, ele oferece maior taxa de processamento (*throughput*), geração de tokens mais rápida e melhor desempenho em benchmarks comuns em comparação com os modelos Flash anteriores.

O DeepSeek-V3.2-Exp[3] é um modelo de linguagem de grande porte experimental lançado pela DeepSeek como um passo intermediário entre o V3.1 e futuras arquiteturas. Ele foi projetado para melhorar a eficiência de treinamento e inferência em cenários de longo contexto, mantendo a qualidade das respostas.

O Grok 4 Fast[4] é o modelo multimodal mais recente da xAI, oferecendo eficiência de custo e uma janela de contexto de 2 milhões de tokens.

O Llama 3.1[5] tem 70 bilhões de parâmetros e é otimizado para casos de uso que envolvem diálogos de alta qualidade. Segundo o provedor, este modelo demonstrou desempenho robusto em comparação com os principais modelos de código fechado em avaliações conduzidas com participantes humanos.

---

1 https://openrouter.ai/openai/gpt-oss-20b:free
2 https://openrouter.ai/google/gemini-2.5-flash-lite
3 https://openrouter.ai/deepseek/deepseek-v3.2-exp
4 https://openrouter.ai/x-ai/grok-4-fast
5 https://openrouter.ai/meta-llama/llama-3.1-70b-instruct

O GPT-5 Mini[6] é o sucessor do modelo o4-mini da OpenAI e uma versão compacta do GPT-5, projetada para lidar com tarefas de raciocínio mais leves. Ele oferece os mesmos benefícios de seguimento de instruções e ajustes de segurança do GPT-5, mas com latência e custo reduzidos.

Tabela 1 – Características dos Modelos de LLM Avaliados

Atributo	**OpenAI GPT Oss**	**Gemini 2.5 Flash**	**DeepSeek 3.2**	**Grok4 Fast**	**Llama 3.1**	**GPT5 Mini**
Data criação	Ago/25	Set/25	Set/25	Set/25	Jul/24	Ago/25
Tam. Contexto	131K	1.05M	164K	2M	131K	400K
Max. Saída	131K	66K	163.8K	30K	131K	128K
Custo input U$ (/M tokens)	0,03	0,10	0,27	0,20	0,40	0,25
Custo output U$ (/ M tokens)	0,14	0,40	0,40	0,50	0,40	2,0
Formatos de entradas	texto	arquivo, imagem, texto, áudio	texto	texto, imagem	texto	texto, imagem, arquivo
Formatos de Saída	texto	texto	texto	texto	texto	texto
Capacidade avançada de raciocínio	sim	sim	sim	sim	não	sim

Foram feitas tentativas de uso de modelos gratuitos, mas esses possuem limitação de número de requisições diárias e mensais, o que se tornou proibitivo para a estratégia de validação proposta. Dessa forma, todos os modelos selecionados são pagos e a escolha buscou seguir uma lógica de compromisso entre custo e capacidade. Outro fator considerado na escolha foi o compromisso dos provedores de não utilizar os prompts para treinamento e evolução do modelo. Embora a solução não lide com dados sensíveis de pacientes, o material de referência é uma curadoria do CIM-POA e representa informação proprietária da organização.

Uma das características consideradas relevantes para esse trabalho é o tamanho do contexto de entrada e saída oferecido pelo modelo. Em princípio, o problema não exige grande capacidade nesse quesito pois as perguntas e respostas são objetivas, o material de referência não é extenso e apenas os 3 documentos mais relevantes são incluídos como contexto no prompt. O Gemini 2.5 Flash e o Grok4 Fast são os modelos com maior

[6] https://openrouter.ai/openai/gpt-5-mini

limitação de contexto de saída e possuem um custo similar. Já o Llama 3.1 e o OpenAI GPT OSS possuem maior limitação de tamanho do contexto total (entrada e saída), mas apresentam grande diferença no custo. O Llama 3.1 é o único modelo lançado há mais de 1 ano e que não oferece capacidade avançada de raciocínio, embora seja uma das alternativas de maior custo. Por fim, tanto os prompts quanto as saídas esperadas são totalmente textuais, sem informações adicionais em outros formatos. Assim, foi dada preferência para modelos que sejam focados nesse tipo de entrada e saída. Os modelos GPT5 Mini e Gemini 2.5 Flash permitem, respectivamente, a entrada em arquivo e áudio. Embora ainda não utilizadas nesta PoC, essas opções podem ser úteis para a solução aprimorada, em especial a entrada por áudio.

A Tabela 2 resume os *scores* de similaridade informados por cada modelo quando o mesmo LLM é usado para gerar e validar a resposta a uma consulta.

Tabela 2 – Desempenho dos modelos selecionados (LLM de geração é o mesmo de validação)

Medida	**OpenAI GPT OSS**	**Gemini 2.5 Flash**	**DeepSeek 3.2**	**Grok 4 Fast**	**Llama 3.1**	**GPT5 Mini**
Média	50.91	53.62	53.75	51.07	40.95	49.95
Mediana	60	57.5	57.5	45	35	42.5
Desvio Padrão	33.22	26.47	27.52	22.39	32.75	24.21
Mínimo	0	5	0	20	0	20
Máximo	99	98	100	90	95	95
Q1 (25%)	10	30	30	32.5	9.25	30
Q3 (75%)	80	75	75	70	70	71.25
IQR (intervalo interquartil)	70	45	45	37.5	60.75	41.25
p-valor normalidade	8.74E-08	7.78E-03	6.75E-04	2.00E-04	4.23E-09	6.30E-04

A título de exemplo, as Figuras 5 e 6 mostram os histogramas para os modelos DeepSeek e Llama, que apresentaram, respectivamente, o melhor e o pior *score* médio.

A Tabela 3 resume os resultados obtidos por cada modelo quando um LLM diferente é usado para validar a resposta. Para este experimento, o modelo Llama foi usado para validar as respostas dos demais modelos e o GPT5 Mini validou as respostas geradas pelo Llama.

As Figuras 7 e 8 mostram os histogramas para os modelos DeepSeek e Gemini, que apresentaram, respectivamente, o melhor e o pior *score* médio quando o Llama foi usado como mecanismo de validação.

Pode-se observar na Tabela 2 que todos os modelos apresentaram uma média baixa de *scores* de similaridade. Nenhum modelo se sobressaiu de maneira evidente e o Llama

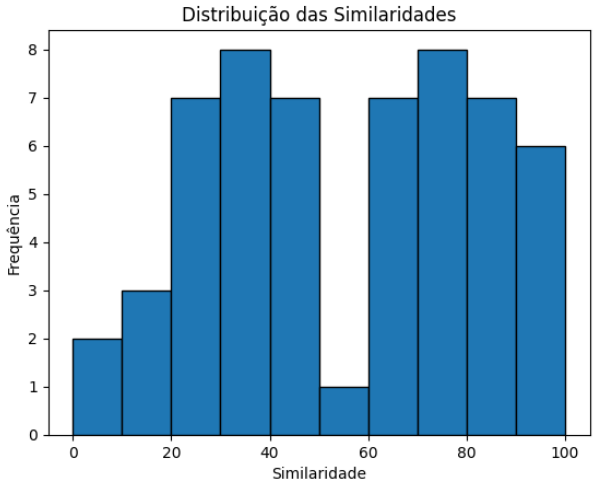

Figura 5 – Distribuição do *score* no modelo DeepSeek 3.2

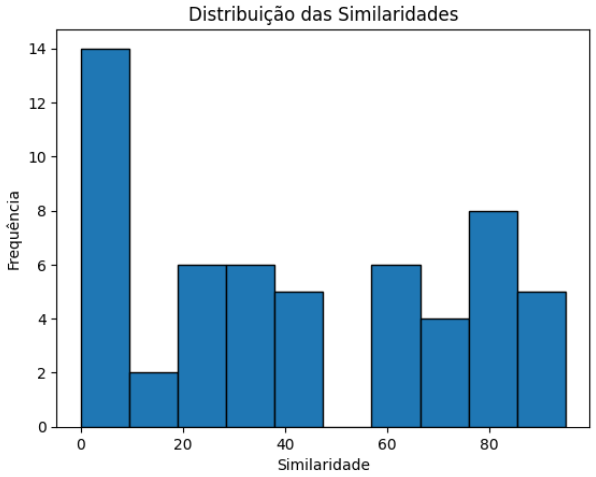

Figura 6 – Distribuição do *score* no modelo Llama 3.1

3.1, especificamente, apresentou um resultado bem pior (score de similaridade de 41% em média). As Figuras 5 e 6 demonstram visualmente a grande variação da qualidade da resposta, conforme indicado pelos valores de desvio padrão. O GPT-5 Mini, embora seja o modelo mais custoso, não apresenta o melhor resultado nem maior consistência. O modelo mais barato (GPT OSS) não é o de pior desempenho, mas apresenta a maior dispersão.

A Tabela 3 apresenta os resultados quando LLMs distintos são usados para geração e validação da resposta. Os *scores* médios continuaram baixos e um pouco menores do que os apresentados na Tabela 2 em todos os casos (em média 1 ponto percentual abaixo). O DeepSeek manteve o melhor desempenho, mas o pior resultado passou a ser do Gemini. O resultado, mais uma vez, parece não depender do custo do modelo.

Como demonstram os histogramas apresentados e os valores de p-normalidade, a distribuição dos *scores* de similaridade em todos os modelos é não normal, ou seja, o desempenho de todos os modelos é não homogêneo, o que dificulta a comparação entre eles.

Tabela 3 – Desempenho dos modelos selecionados (LLM de geração é diferente do modelo de validação)

Medida	OpenAI GPT OSS	Gemini 2.5 Flash	DeepSeek 3.2	Grok 4 Fast	Llama 3.1	GPT5 Mini
Média	45.23	42.75	48.48	46.96	47.73	44.73
Mediana	47.5	40	40	40	40	40
Desvio Padrão	33.26	28.31	29.13	28.23	27.83	29.37
Mínimo	0	0	0	0	0	0
Máximo	95	97	95	95	98	90
Q1 (25%)	9.5	20	20	20	25	20
Q3 (75%)	80	62.5	80	80	75	80
IQR (intervalo interquartil)	70.5	42.5	60	60	50	60
p-valor normalidade	4.89E-10	1.20E-03	4.38E-05	5.26E-05	5.73E-04	2.81E-09

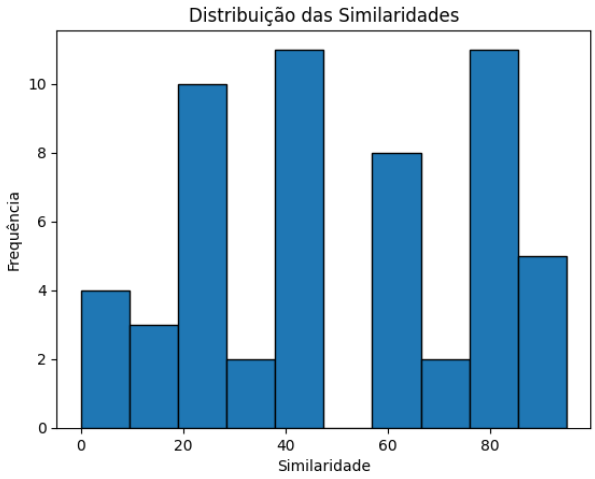

Figura 7 – Distribuição do *score* no modelo DeepSeek 3.2

Tabela 4 – Resultados do Teste de Kruskal-Wallis

Medida	Mesma LLM para geração e validação	LLMs distintas para geração e validação
Estatística H de Kruskal-Wallis	7,18	1,75
p-valor	0,208	0,882

Assim, foi feito o teste de Kruskal-Wallis para avaliar se algum modelo se destaca, ainda que o resultado absoluto esteja abaixo do esperado. Os resultados desse teste para as duas formas de validação (um único modelo para geração e validação ou modelos distintos de geração e validação) são apresentados na Tabela 4 e confirmam que não houve diferença estatisticamente significativa entre os modelos em nenhum caso.

Os resultados mostram que o desempenho médio obtido pela implementação

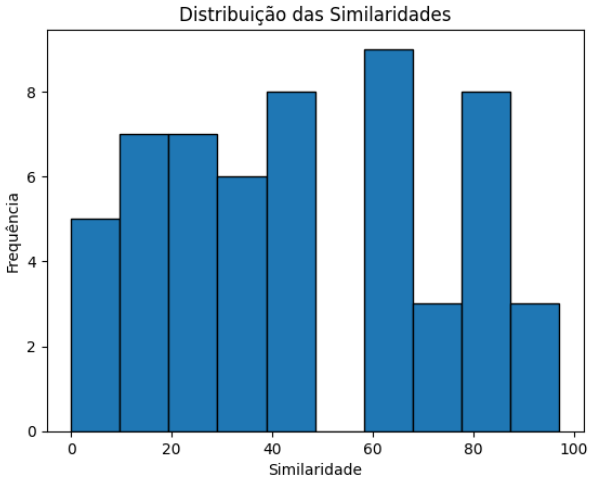

Figura 8 – Distribuição do *score* no modelo Llama 3.1

proposta é insuficiente para o domínio da aplicação, não conferindo confiabilidade suficiente. Para entender um pouco melhor as causas desse baixo desempenho, foi feita uma inspeção manual nas respostas geradas durante a etapa de validação. Foi possível observar em todos os experimentos uma taxa relativamente alta de respostas do tipo ”Não sei” ou ”Não foi possível encontrar...”. Como mostrado na Figura 2, o LLM é instruído a responder dessa forma quando não encontra subsídios para a resposta no contexto passado como parâmetro. Por exemplo, foram encontradas 34 ocorrências do termo ”Não sei” e similares nas respostas obtidas com o modelo GeminiFlash e 50 ocorrências nos resultados do GPT5 Mini. Cada experimento faz 56 consultas. Ainda que os termos mencionados possam se repetir em uma mesma resposta, os valores encontrados permitem levantar a hipótese de que o contexto fornecido para o RAG não é suficiente.

Assim, em relação às hipóteses levantadas, pode-se concluir que:

Hipótese 1: Não foi validada pela solução implementada, pois os *scores* de similaridade ficaram muito abaixo da taxa de 90% estabelecida. Os resultados também não permitem rejeitar definitivamente essa hipótese pois são necessários mais experimentos explorando outros parâmetros da solução que não foram totalmente explorados, como, por exemplo, o tamanho do contexto passado junto com o prompt.

Hipótese 2: Não foi possível distinguir o desempenho entre os modelos de maior e menor custo considerando as medidas utilizadas. Portanto, essa hipótese não foi validada pela implementação feita.

5.2 Ameaças à Validade

As seguintes limitações foram identificadas para este estudo:

5.2.1 Validade interna

Em relação à validade interna, as principais ameaças identificadas estão relacionadas ao modelo de *embedding*, ao dataset e à métrica utilizados na avaliação.

O cálculo da similaridade de cosseno utilizado na identificação dos documentos de referência depende do modelo de *embedding* escolhido e os experimentos foram feitos utilizando um único modelo. É possível que o modelo escolhido não capture bem diferenças semânticas importantes, distorcendo a comparação. Para mitigar este risco, foi escolhido um modelo bem referenciado e voltado a múltiplas línguas, conforme uma característica do dataset. No entanto, considerando a baixa qualidade dos resultados obtidos, esse é um parâmetro importante a ser explorado em trabalhos subsequentes.

O dataset de avaliação usa apenas 56 pares de perguntas e respostas provenientes do CIM-POA e há risco de viés de amostragem se esse conjunto não for representativo da variedade de perguntas reais que o sistema enfrentará. Considerou-se esse risco baixo uma vez que o dataset representa solicitações reais feitas ao CIM-POA no período de 2 meses.

A avaliação se baseou apenas no *score* de similaridade que foi definido pelo LLM. O uso de um LLM distinto na etapa de validação buscou mitigar o risco de viés de confirmação, mas nenhuma outra providência foi tomada em relação ao risco de alucinação, por exemplo. Além disso, não foi feita ainda uma validação manual das respostas obtidas e essa é uma etapa fundamental para avaliar inclusive quais parâmetros devem ser alterados para melhorar o desempenho da solução.

5.2.2 Validade de construto

O conceito de eficácia de um agente não é definido por um único critério. Modelos podem gerar respostas corretas em termos de conteúdo, mas com linguagem diferente. O uso de LLMs na etapa de validação teve com objetivo lidar com essa questão. No entanto, não foram feitos ajustes no contexto associado a cada consulta. Além disso, como mencionado, as respostas oferecidas pelos modelos não foram ainda avaliadas pelos profissionais envolvidos na operação do CIM-POA. Esse tipo de validação é custoso e o objetivo desse trabalho era gerar uma PoC de uma solução automatizada e obter uma primeira avaliação do potencial dessa solução. Embora a confiabilidade pela métrica utilizada não tenha sido satisfatória, ela ficou ainda em patamares que justificam um estudo mais aprofundado dos demais parâmetros (modelos de *embedding*, configurações dos LLMs, definições do RAG e do prompt, entre outros) visando melhorar o desempenho da solução antes de assumir o custo de uma validação humana.

5.2.3 Ameaças à confiabilidade

LLMs são, por definição, não determinísticas e podem gerar respostas incorretas mesmo quando orientadas a utilizar referências específicas. Uma boa prática é a realização

de múltiplas execuções da mesma demanda juntamente com o controle de parâmetros como temperatura de forma que variações entre execuções sejam também consideradas. Essa prática não foi aplicada nos experimentos apresentados por duas razões: i)limitação de tempo e recursos e ii) prioridade de análise. Como mencionado, foram utilizados modelos pagos para que a demanda de requisições da avaliação fosse atendida. Além disso, alguns modelos apresentaram execuções mais lentas para o dataset de avaliação. O foco dessa primeira análise era avaliar o potencial da solução e, eventualmente, identificar um modelo mais adequado à tarefa proposta. Por essas razões, cada modelo foi executado uma única vez. A partir da identificação de um modelo que seja considerado o mais adequado, uma avaliação mais detalhada deve ser feita para que seja medido também o grau de não determinismo e quais medidas devem ser tomadas na solução final para garantir a confiabilidade das respostas.

Em relação à validade externa, não foram identificadas limitações pertinentes a esse estudo.

6 CONCLUSÕES

Este trabalho apresentou uma PoC para automatizar o serviço de informação ativa em um CIM, visando reduzir a sobrecarga operacional da equipe e o comprometimento do tempo de resposta. A solução proposta consistiu em um QAS que utiliza a arquitetura RAG em conjunto com LLMs. Documentos de referência do CIM foram usados como dataset de referência para o RAG e a geração das respostas às consultas foi feita usando LLMs sob a persona de um farmacêutico clínico, com a instrução de usar apenas o contexto fornecido. A avaliação de desempenho da solução foi realizada com 56 pares de perguntas e respostas reais e utilizando um LLM distinto para validar a similaridade semântica das respostas geradas com as esperadas.

A PoC implementada neste trabalho cumpriu integralmente o objetivo específico de entender o processo de criação de uma solução com base em LLMs e RAG, ao detalhar e executar cada fase da arquitetura proposta, desde a ingestão e indexação dos documentos de referência especializada do CIM-POA até a lógica de Recuperação e Geração de Respostas. Em especial, foi possível compreender os desafios para a obtenção de dados relevantes para um problema, aprimorar o processo de estruturação de uma solução usando tecnologias de ponta e experimentar algumas técnicas e tecnologias já disponíveis no mercado.

O objetivo de avaliar o desempenho de diferentes modelos de LLM em termos de custo e qualidade da resposta foi alcançado por meio da estratégia de validação que submeteu 56 consultas reais a seis modelos de LLM distintos e mediu a similaridade semântica das respostas geradas. Os resultados experimentais indicaram um desempenho médio insuficiente para a confiabilidade esperada e não mostraram diferença estatisticamente significativa entre os modelos testados. A partir de uma análise qualitativa, foram identificados alguns parâmetros que podem e/ou devem ser ajustados nesse tipo de solução para alcançar a confiabilidade desejada. Um grande número de respostas apontaram para a ausência de subsídios para o esclarecimento da consulta feita. Isso sugere que parâmetros do RAG, como a suficiência do contexto recuperado e o modelo de *embedding* utilizado, podem ser o foco de futuros experimentos para que a confiabilidade requerida em um ambiente clínico-hospitalar possa ser atingida. Um aprendizado importante derivado desse objetivo foi um melhor entendimento sobre as reais limitações dos modelos LLM e a necessidade de um ajuste fino para que resultados confiáveis sejam atingidos.

Os baixos *scores* de similaridade e a alta taxa de respostas "Não sei" indicam que o contexto fornecido ao RAG pode ser insuficiente. Trabalhos futuros devem focar em identificar e ajustar os parâmetros críticos da solução (como o modelo de embedding escolhido, o tamanho dos *chunks*, a quantidade de contexto no prompt ou mesmo o uso de estratégias mais elaboradas de RAG) antes de buscar uma validação humana. Além disso,

a conclusão de que não houve diferença estatisticamente significativa no desempenho entre modelos de LLM de alto e baixo custo sugere que o CIM pode não precisar investir nos modelos mais caros para o problema proposto, desde que os parâmetros de RAG sejam otimizados. Dessa forma, o foco de investigação deve se manter na engenharia do sistema e não apenas na capacidade bruta do LLM. Por fim, o projeto deve ainda considerar as implicações de segurança e privacidade dos dados uma vez que o material de referência é proprietário e eventualmente podem surgir consultas que exponham dados de pacientes. Nesse sentido, o uso de LLMs instalados localmente deve ser também investigado.

REFERÊNCIAS

ANANDARAJAN, M.; HILL, C.; NOLAN, T. Introduction to text analytics. *In*: _____. **Practical Text Analytics: Maximizing the Value of Text Data**. Cham: Springer International Publishing, 2019. cap. 1, p. 1–11. ISBN 978-3-319-95663-3. Disponível em: https://doi.org/10.1007/978-3-319-95663-3_1.

BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern information retrieval : the concepts and technology behind search**. 2. ed. New York: Addison Wesley, 2011. 913 p. ISBN 9780321416919, 0321416910.

BIANCOFIORE, G. M. *et al.* Interactive question answering systems: Literature review. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 56, n. 9, maio 2024. ISSN 0360-0300. Disponível em: https://doi.org/10.1145/3657631.

BROWN, T. B. *et al.* Language models are few-shot learners. *In*: **Proceedings of the 34th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546.

BRUNIALTI, L. *et al.* Aprendizado de máquina em sistemas de recomendação baseados em conteúdo textual: Uma revisão sistemática. *In*: **Anais do XI Simpósio Brasileiro de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2015. p. 203–210. ISSN 0000-0000. Disponível em: https://sol.sbc.org.br/index.php/sbsi/article/view/5818.

EISENSTEIN, J. **Introduction to Natural Language Processing**. 1. ed. Cambridge: MIT Press, 2019. 536 p. ISBN 9780262042840.

ESTRATéGICOS, D. de Assistência Farmacêutica e I. **Centros e Serviços de Informação sobre Medicamentos : princípios, organização, prática e trabalho em redes para promoção do Uso Racional de Medicamentos**. Brasília: Ministério da Saúde, 2020. 251 p. ISBN 978-85-334-2768-6. Disponível em: http://bvsms.saude.gov.br/publicacoes/centros_servicos_informacao_medicamentos.pdf.

INC., M. **Meta Llama 3**. 2025. Disponível em: https://about.fb.com/br/news/2024/04/apresentando-meta-llama-3-o-grande-modelo-de-linguagem-de-codigo-aberto-mais-capaz-ate-hoje/.

IZACARD, G.; GRAVE, E. Leveraging passage retrieval with generative models for open domain question answering. *In*: MERLO, P.; TIEDEMANN, J.; TSARFATY, R. (ed.). **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**. Association for Computational Linguistics, 2021. p. 874–880. Disponível em: https://aclanthology.org/2021.eacl-main.74/. Acesso em: 25 out. 2025.

JAKUS, G. *et al.* **Concepts, Ontologies, and Knowledge Representation**. 1. ed. New York: Springer New York, NY, 2013. 67 p. ISBN 978-1-4614-7822-5.

KAPLAN, J. *et al.* **Scaling Laws for Neural Language Models**. 2020. Disponível em: https://arxiv.org/abs/2001.08361.

KHURANA, D. *et al.* Natural language processing: state of the art, current trends and challenges. **Multimedia Tools and Applications**, Porto Alegre, v. 82, n. 3, p. 3713–3744, 2023.

LEWIS, P. *et al.* Retrieval-augmented generation for knowledge-intensive nlp tasks. *In*: **Proceedings of the 34th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2020. (NIPS '20). ISBN 9781713829546.

LIU, P. *et al.* Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 9, jan. 2023. ISSN 0360-0300. Disponível em: https://doi.org/10.1145/3560815.

M. ZHANG W.E., S. Q. e. a. Z. Conversational question answering: a survey. **Knowledge and Information Systems**, Springer Nature, v. 64, p. 3151–3195, december 2022.

MITCHELL, T. M. **Machine Learning**. 1. ed. New York: McGraw-Hill, 1997. 432 p. ISBN 0070428077.

MOLLá-ALIOD, D.; VICEDO, J.-L. Question answering. *In*: INDURKHYA, N.; DAMERAU, F. (ed.). **Handbook of Natural Language Processing**. [*S.l.: s.n.*]: CRC Press, 2010, (Chapman & Hall/CRC Machine Learning & Pattern Recognition). cap. 20. ISBN 9781420085938.

OPENAI. **ChatGPT**. 2025. Disponível em: https://chatgpt.com/.

RAM, O. *et al.* In-context retrieval-augmented language models. **Transactions of the Association for Computational Linguistics**, MIT Press, Cambridge, MA, v. 11, p. 1316–1331, 2023. Disponível em: https://aclanthology.org/2023.tacl-1.75/.

REIMERS, N.; GUREVYCH, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *In*: **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing**. Association for Computational Linguistics, 2019. Disponível em: http://arxiv.org/abs/1908.10084.

REZENDE, S. O. (ed.). **Sistemas Inteligentes: fundamentos e aplicações**. 1. ed. Barueri: Manole, 2003. 525 p. ISBN 85-204-1683-7.

SHANAHAN, M.; MCDONELL, K.; REYNOLDS, L. Role play with large language models. **Nature**, Nature Publishing Group UK London, v. 623, n. 7987, p. 493–498, 2023.

SILVA, R.; GOMES, L. An adaptive language model-based intelligent medication assistant for the decision support of antidepressant prescriptions. **Computers in Biology and Medicine**, v. 190, p. 110065, 2025. ISSN 0010-4825. Disponível em: https://www.sciencedirect.com/science/article/pii/S0010482525004160.

VASWANI, A. *et al.* **Attention Is All You Need**. 2023. Disponível em: https://arxiv.org/abs/1706.03762.

WEI, J. *et al.* Chain-of-thought prompting elicits reasoning in large language models. *In*: **Proceedings of the 36th International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2022. (NIPS '22). ISBN 9781713871088.

WHITE, J. *et al.* A prompt pattern catalog to enhance prompt engineering with chatgpt. *In*: **Proceedings of the 30th Conference on Pattern Languages of Programs**. USA: The Hillside Group, 2023. (PLoP '23). ISBN 9781941652190.

ZHANG, S. *et al.* Instruction tuning for large language models: A survey. **ArXiv**, abs/2308.10792, 2023. Disponível em: https://api.semanticscholar.org/CorpusID:261049152.

ZHAO, Z. *et al.* Recommender systems in the era of large language models (llms). **IEEE Transactions on Knowledge and Data Engineering**, v. 36, n. 11, p. 6889–6907, 2024.

ZHOU, Q. *et al.* Clinicalrag: Automating pharmaceutical label quality control with hierarchical rag and large language models. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 39, n. 28, p. 29736–29738, Apr. 2025. Disponível em: https://ojs.aaai.org/index.php/AAAI/article/view/35384.